N94-35064

# Mission Evaluation Room Intelligent Diagnostic and Analysis System (MIDAS)


Author: Ginger L. Pack/ER221

Position/Title: MIDAS Project Manager

Affiliation: NASA (ER22)
Address: Lyndon B. Johnson Space Center
Houston, TX 77058

Phone: (713) 483-1515
FAX : (713) 483-3204

Co-Authors:
1. Jane Falgout
2. Joseph Barcio
3. Steve Shnurer
4. David Wadsworth
5. Louis Flores

Affiliation: Lockheed Engineering & Sciences Corporation

Address: ESPO C19
PO Box 58561

Phone : (713) 483-2056



## ABSTRACT

The role of Mission Evaluation Room (MER) engineers is to provide engineering support during Space Shuttle missions, for Space Shuttle systems. These engineers are concerned with ensuring that the systems for which they are responsible function reliably, and as intended. The MER is a central facility from which engineers may work, in fulfilling this obligation. Engineers participate in real-time monitoring of shuttle telemetry data and provide a variety of analyses associated with the operation of the shuttle.

The Johnson Space Center's Automation and Robotics Division is working to transfer advances in intelligent systems technology to NASA's operational environment. Specifically, the MER Intelligent Diagnostic and Analysis System, (MIDAS) project provides MER engineers with software to assist them with monitoring, filtering and analyzing Shuttle telemetry data, during and after Shuttle missions. MIDAS off-loads to computers and software, the tasks of data gathering, filtering, and analysis, and provides the engineers with information which is in a more concise and usable form needed to support decision making and engineering evaluation. Engineers are then able to concentrate on more difficult problems as they arise.

This paper describes some, but not all of the applications that have been developed for MER engineers, under the MIDAS Project. The sampling described herewith was selected to show the range of tasks that engineers must perform for mission support, and to show the various levels of automation that have been applied to assist their efforts.


## INTRODUCTION

The purpose of the MIDAS project is to provide MER engineers with real time intelligent software to assist them with monitoring, filtering and analyzing Shuttle telemetry data, during and after Shuttle missions, and to capture the expertise held by highly experienced engineers. This is accomplished by applying advanced automation and intelligent systems to the tasks required to perform engineering evaluation.

The MER engineers perform a variety of task in support of Shuttle missions. They continuously monitor and evaluate the performance and health of their systems, provide engineering analysis

support to Mission Operations personnel, and are responsible for certifying the integrity of orbiter systems for subsequent flights. Historically, engineers have had to rely upon several screens full of telemetry data, displayed as ASCII text and numerical values, to understand the state and operating condition of their system. Information in this form is crammed onto displays that engineers must monitor and interpret, (Figure 1). Several of these displays exist for each subsystem. Engineers visually monitor these displays during missions, to obtain an understanding of the behavior of their systems. Only one display at a time may be shown. This arrangement precludes the engineer from noticing changes in information that are not contained on the display being viewed. The selection of screens available for viewing is controlled by Mission Operations personnel. Consequently, during missions engineers must often "chase around" the displays which contain the information in which they are interested. This further complicates the duties that engineers must perform, in maintaining the health of the systems for which they are responsible. Additionally, engineers use manual methods to perform analysis. Events are logged by hand, and associated analysis is performed using pencil and paper. This mode of operation can be tedious, time consuming, and vulnerable to the errors that result from distraction and boredom.

## TECHNICAL APPROACH

The first task to be done in automating the MER, was to analyze the current methods of operating, and identify problems and deficiencies. A questionnaire was developed to learn where engineers were spending their time, which tasks were labor intensive, boring, repetitive, and just hard to do within current processes and methods. Discussion of these issues involved knowledge engineering to capture system expertise, and determine the kinds of analysis with which engineers needed help. It was learned that engineers wanted assistance in recognizing trends that were reflected in the data, and evaluating the potential impact on hardware. They also needed to capture the data and perform identification and diagnosis of problems before a critical state is reached. Additionally, engineers wanted help in capturing and analyzing in-flight data to check out and verify the health of the systems for future flights. Finally, they needed relief from the task of visually monitoring and manually gathering the data needed to perform various analyses. Computers and software could provide for these needs, and enable the engineers to focus their efforts on problems as they arise versus investing so much effort gathering and sifting through large amounts of data to evaluate system health and performance. Using the questionnaire as a guide, developers work with engineers to identify troublesome areas and to reach agreement on which tasks are to be built into software.

Automation requirements are developed through close interaction with Shuttle subsystem engineers. User feedback is solicited and incorporated at every phase of product definition and development. Developers work closely with users while developing the architectural design, and while the implementing the design. Frequent product reviews are also conducted with the user to ensure appropriateness and accuracy of results. User involvement in all phases of development results in products that are familiar to users and fit well to the user's needs. It also ensures user "buy in" to the technology, since they help to define the implementation of the technological solution.

## ACCOMPLISHMENTS

### Discrete Log

The Discrete Monitor and Logging program (figure 2), was created to relieve engineers of the continuous eyes of monitoring on system data. Previously, this task was done by having a person watch a screen full of data (figure 1), noting the occurrence of certain events, and writing them down on a sheet of paper. The Discrete Monitor and Logging program reads the shuttle telemetry stream and automatically detects changes in the state of discrete parameters. The changes are

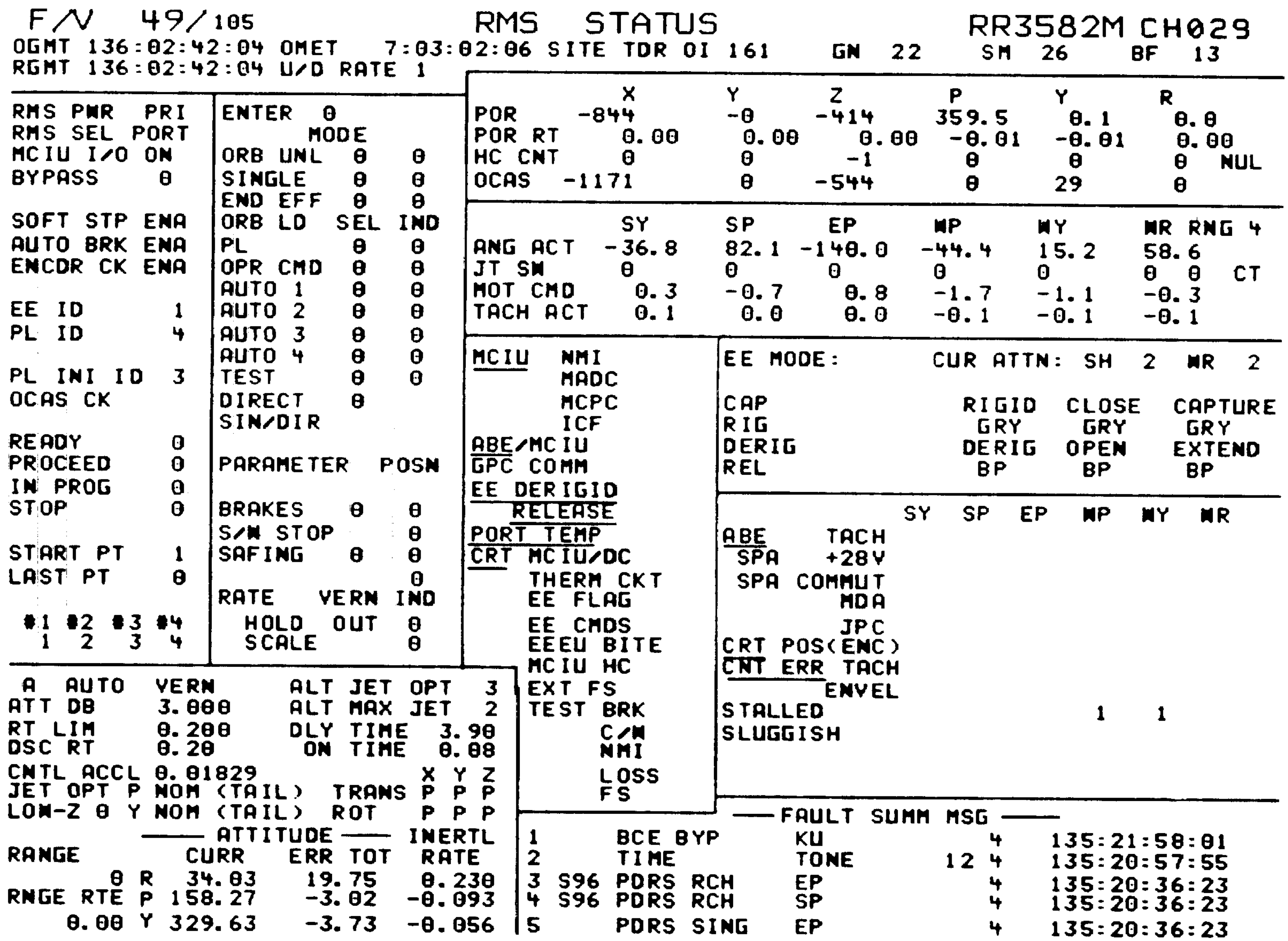


Figure 1: Data Display

## Discrete Log

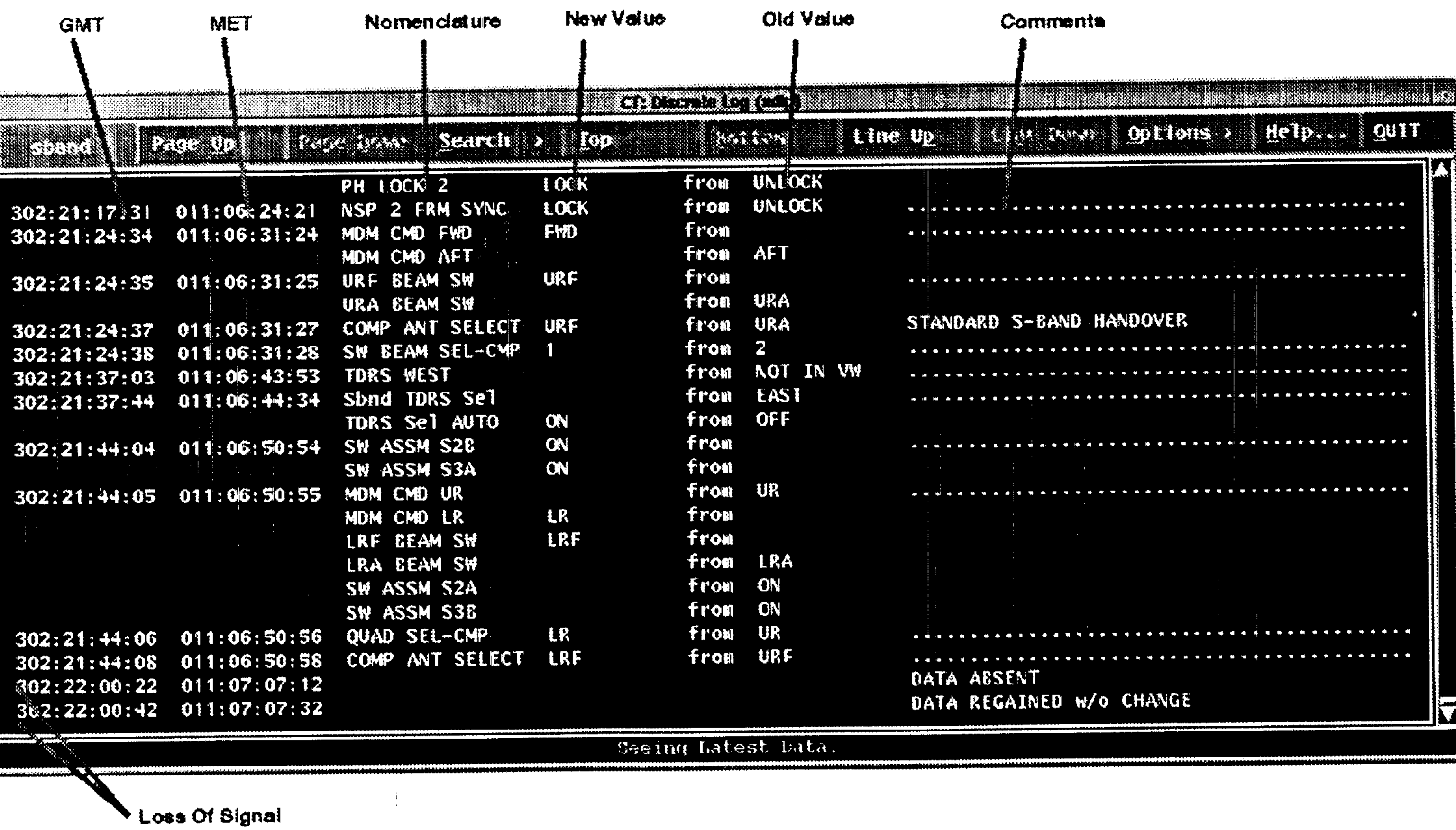


Figure 2: Discrete Monitor & Logging Application

logged and written to the screen, and to a file that may be translated to ASCII format for further manipulation by the user. The entries consist of a description of the changed event, the Greenwich Mean Time (GET) and Mission Elapsed Time (MET) of the change, the previous state, the current state, and a comment field. The event descriptor and state changes are entered in English. Comments may be entered from a template or edited by the user. The log entries are painted yellow when data has changed during a period of signal loss. The first version of the log, logged each change without regard to how long or how often the parameter had changed or was changing. Consequently, parameters that "fluttered" on and off quickly filled up the log. This was taken care of by adding a module that would wait for the data to stabilize before reporting that the data had changed.

### Analog Plot

A companion application to the Discrete Monitor and Logging program is the Analog Plot program (figure 3). The Analog Plot program graphically depicts system behavior that is described by analog values. The plots appear as real time strip charts but are defined by conditional logic among the analog values. Thirty minutes of data are shown. The plots are scrolled every ten minutes. Plots are organized in "families" that are made up of pages. These pages can be combined in mix and match fashion to facilitate visual comparison of data. A "snap shot" of interesting data can also be saved and re-loaded for comparison against other data.

### Ku-band Automated Self-test Analyst

The Ku-band Automated Self-Test Analyst (KASTA) is an expert system that detects the start of the Ku-band self-test, and interprets the test results. The self test is done prior to deployment of the shuttle's Ku-band antenna. The purpose of the self-test is to ensure that the Ku-band antenna is healthy and will operate correctly when used. The self-test consist of several sub-tests whose outcome depends on the success or failure of previous tests. The pass/fail condition of different combinations determine the success or failure of the entire Ku-band self-test, and consequently whether the antenna is fit for use. The expertise for analyzing this test was held by the Ku-band subsystem manager, requiring her to be available for every Ku-band self-test. KASTA allows non-expert personnel to evaluate the self-test.

KASTA evaluates the tests that determine the ability of the Ku-band antenna to achieve proper position within allowed time, verifies the ability of the antenna to track properly, accounts for and manages exceptions to self-test procedure, and recognizes non-significant "failures". KASTA uses telemetry to identify initiation of the self-test, and begins to monitor each test according to pass/fail criteria contained in the knowledge base. Telemetry values are evaluated against rules that determine whether the conditions of the tasks are met, and explanation is provided to support conclusions made by KASTA.

Considerable thought was given to how this information should be conveyed to the user. The application interface provides feedback to the user regarding which task or subtask is in progress, and where in the self test procedure the task falls. Other types of information also had to be communicated through the user interface. The status of each task, task outcome, and the data associated with each task, such as task duration, and parameters checked, had to be presented to the user. Additionally, details concerning the failure of any task to complete had to be communicated. Finally, explanation and supporting data for KASTA's decisions, had to be made available.

The objectives of the display were to strive for easily comprehensible presentation of the various kinds of information, and to provide a highly intuitive display in terms of comprehension of data and in terms of human-computer interaction. These goals were met by grouping like information, arranging information in table format, and using color only to convey specific information. Also, groups of information were placed into windows and labeled using names that clearly described the

## Analog Plots

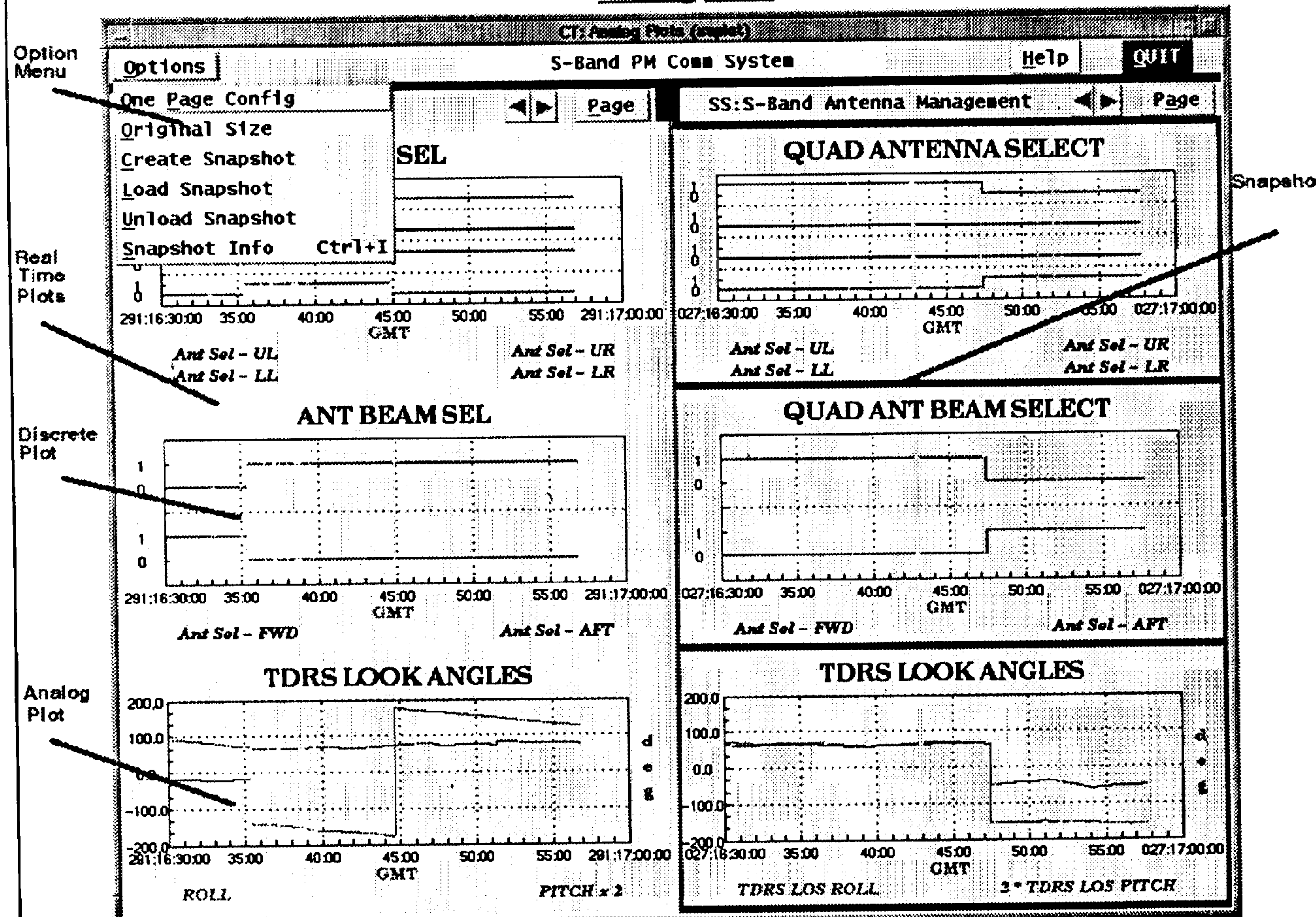


Figure 3: Analog Plot Application

content of the window. The result was a well-organized and highly intuitive display that users could understand and interact with, while having little or no formal instruction (figure 4).

The most important information was placed in the most prominent position of the display area, the center. Supporting data was arranged around the center. The center of the display contains the self test tasks and detailed information pertaining to the tasks. The SELF TEST TASKS window, lists each of the tasks that comprise the Ku-band self-test. The task currently being executed is highlighted in this window. A red or green box indicates whether the task passed (green), or failed (red).

The REAL TIME DATA window is located to the left of the display center, and lists the telemetry and corresponding values that are associated with the self-test. Telemetry corresponding to a task that is in progress, is highlighted while the task proceeds. To the right of the center the SELF TEST STATUS window contains information concerning which task is in progress, time elapsed for the task, start time for the task, and mission time given as Greenwich Mean Time (GMT) and Mission Elapsed Time (MET). Below this window, is a MESSAGES window that allows KASTA to inform the user which tasks have failed.

Users may also review all of the data that KASTA used to determine the outcome of any task. After the self test procedure has completed, the data that was used to determine the outcome of any task is accessible in two ways. The user may review only the data important to a particular task, or review all of the data important to the entire self test procedure.

KASTA performs its analysis by cycling through five distinct phases for every second of telemetry data. The first two phases check the incoming data to prevent KASTA from making decisions using unreliable data. An important benefit of this approach is that KASTA's knowledge base always operates on a time-homogeneous set of data. No analysis rules are allowed to fire until the phase in which data acquisition takes place is finished. This promotes consistency for KASTA's analysis. The five analysis phases are data acquisition, post data acquisition, self test progress update, task progress update, and task outcome analysis.

KASTA's data acquisition strategy first checks the quality of the data before passing the data on. An overall quality indicator determines how well the frame count transmission was received, and is associated with each second's worth of data. If the quality indicator is not 100% that second of data will not be used. An individual quality status indicator is associated with each telemetry value.

During the post data acquisition phase, the telemetry is submitted to more rigorous quality checks. Conditions checked include recency of data, degraded data quality, and bad status associated with the quality indicator or the timestamps, or questionable timestamps associated with the data. If any of these conditions are met for the frame of data (one second of data), none of the remaining phases will be executed. Conversely, if the data passes screening, the next phase of KASTA's analysis is entered. This phase is referred to as the "self test progress phase", and contains the rules that determine when the self test starts and completes. Also included are the rules that determine at any time, how far into the self test procedure, the current analysis is. The third phase of analysis determines how a specific task being executed is progressing. This is the "self test progress update". This phase contains the rules that determine the identity of the current task, and how far into that task the analysis is, at a given time. The outcomes of the tasks are determined during the "task outcome analysis phase". This set of rules specify the pass/fail criteria for the self test, and summarizes the results of all the self test tasks and sub-tasks. KASTA cycles through these phases as long as the self test is in progress. When the self test has completed, the data acquisition and post data acquisition phases will continue to be executed to provide updates to the REAL TIME DATA window.

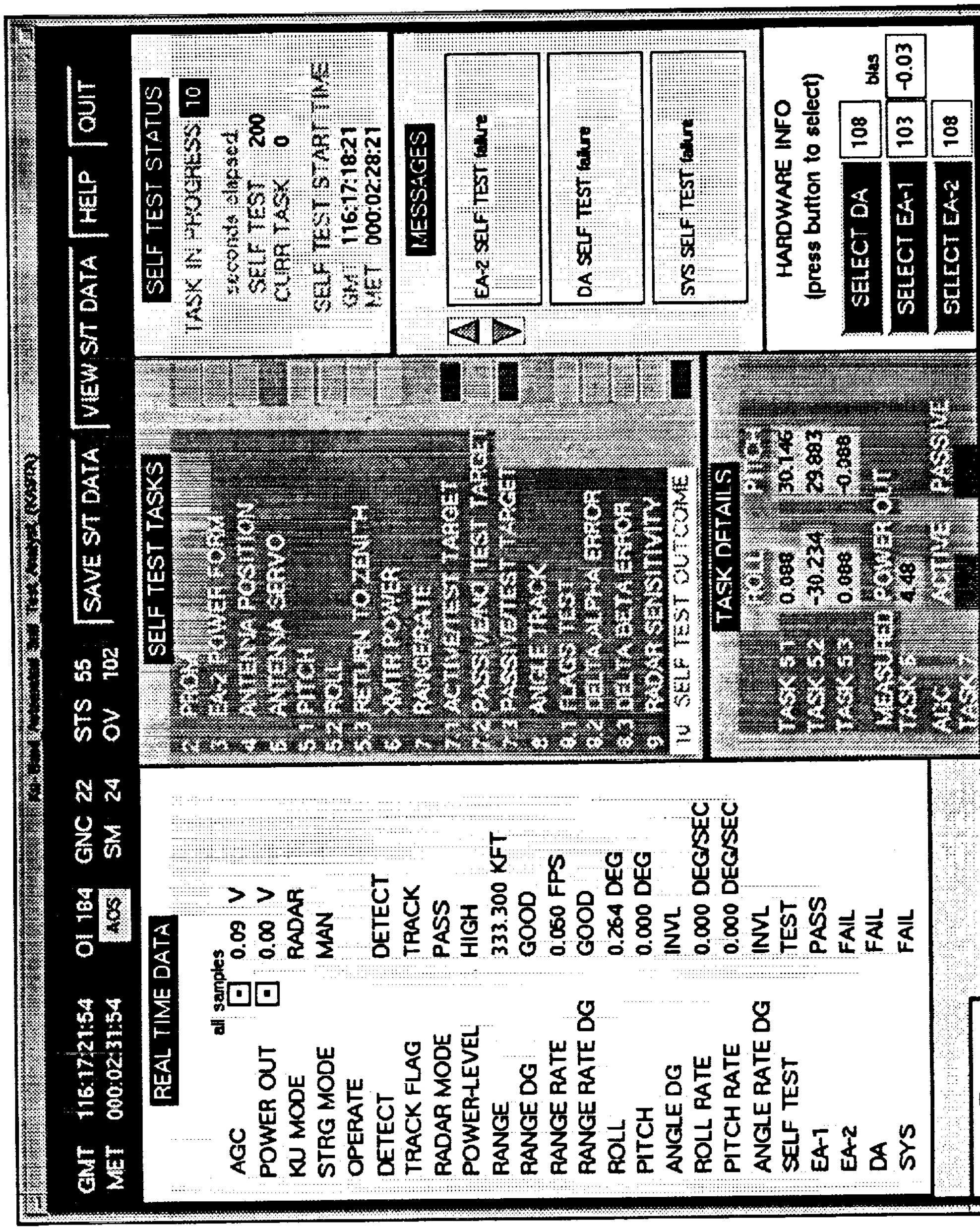


Figure 4: Ku-Band Automated Self Test Analyst (KASTA)

KASTA is written in Gensysms G2 expert system shell. The logic used to analyze the self test in real-time is implemented using a combination of rules, objects, and G2 functions. KASTA is used by MER engineers and, the Integrated Communications Officers (INCOs), who are responsible for Shuttle mission support.

**RMS Monitor**

Since April, 1992, the Remote Manipulator System (RMS) Direct Drive Test (DDT) Monitor has supported missions whenever a flight includes use of the Shuttle robotic arm. The DDT Monitor (figure 5), was constructed to monitor the direct drive test of the RMS joint motors. The six joints of the Shuttle robotic arm are moved by six motors that are individually tested before the arm is deployed. The motors are tested by driving each of them in the forward (positive) and reverse (negative) position. The DDT Monitor automatically detects the initiation of the direct drive test, and plots the results on a screen, as the test occurs. The plot is a visual image of the motor tachometer output, and should produce an expected profile when the motor is operating properly. A warning is issued if the motors are driven beyond 20 seconds a time limit that, if exceeded, invalidates the test. However the application continues to plot the profile. The profile can be saved to an historical file, then retrieved and overlaid against current or other historical data for comparison of current behavior to past behavior. If more that one test is done for a joint motor, the application identifies the subsequent tests and assigns an appropriate number. The plotted image of the test is color coded to correspond with the appropriate number test.

**Automated In-flight Check-out**

The purpose of the in-flight check out is to gather and analyze data needed to verify the health and performance of shuttle systems for subsequent shuttle flights. Performing checkout during flights helps to eliminate the amount of ground testing that must be done between shuttle flights. Existing methods of performing in-flight checkout required engineers to manually gather data throughout the mission, and to pour over large amounts of historical data in search of the information needed to assess and evaluate the in-flight checkout requirements. The Automated In-flight Checkout application automatically gathers the data that is used to conduct in-flight tests, evaluates the data against requirement pass/fail criteria, summarizes the results of the tests, and produces the In-flight Checkout Report. Additionally, when an in-flight checkout requirement is not met, the user is alerted of the discrepancy, and given an opportunity to acknowledge the event and classify it, or to acknowledge it without assigning any classification. The classifications are "failed" for failed requirements that cannot be explained, and "non-problem failure", for failures that can be explained, and require no action. Each reported requirement is associated with a window that when opened, explains what conditions and data caused the test to fail (figure 6).

**PRSD Trend Analysis**

The PRSD Tank Quantity application assists subsystem personnel in detecting leaks in the shuttle oxygen and hydrogen fuel tanks, a problem that has proved to be quite challenging. The application calculates and reports the total accumulated average kilowatt and amperage for the last 24 hours of the mission and, for the accumulated mission time. The H2 and O2 tank quantity differences between known consumption demands and measured depletion is calculated and reported over the most recent 24 hours and for the accumulated mission time. A prediction of time remaining at the current usage rate, and at the average amperage used over the accumulated mission time is calculated and reported. A two day reserve is included in the prediction, and reserve quantities for the H2 and O2 tanks are also reported. (see figure 7)

Theoretically, it should be possible to detect leaks by comparing the measurements provided by tank sensors, to a calculated consumption that can be obtained from the demands of known system loads. However, the sensor reading are affected by several other factors. Side-effects from thruster

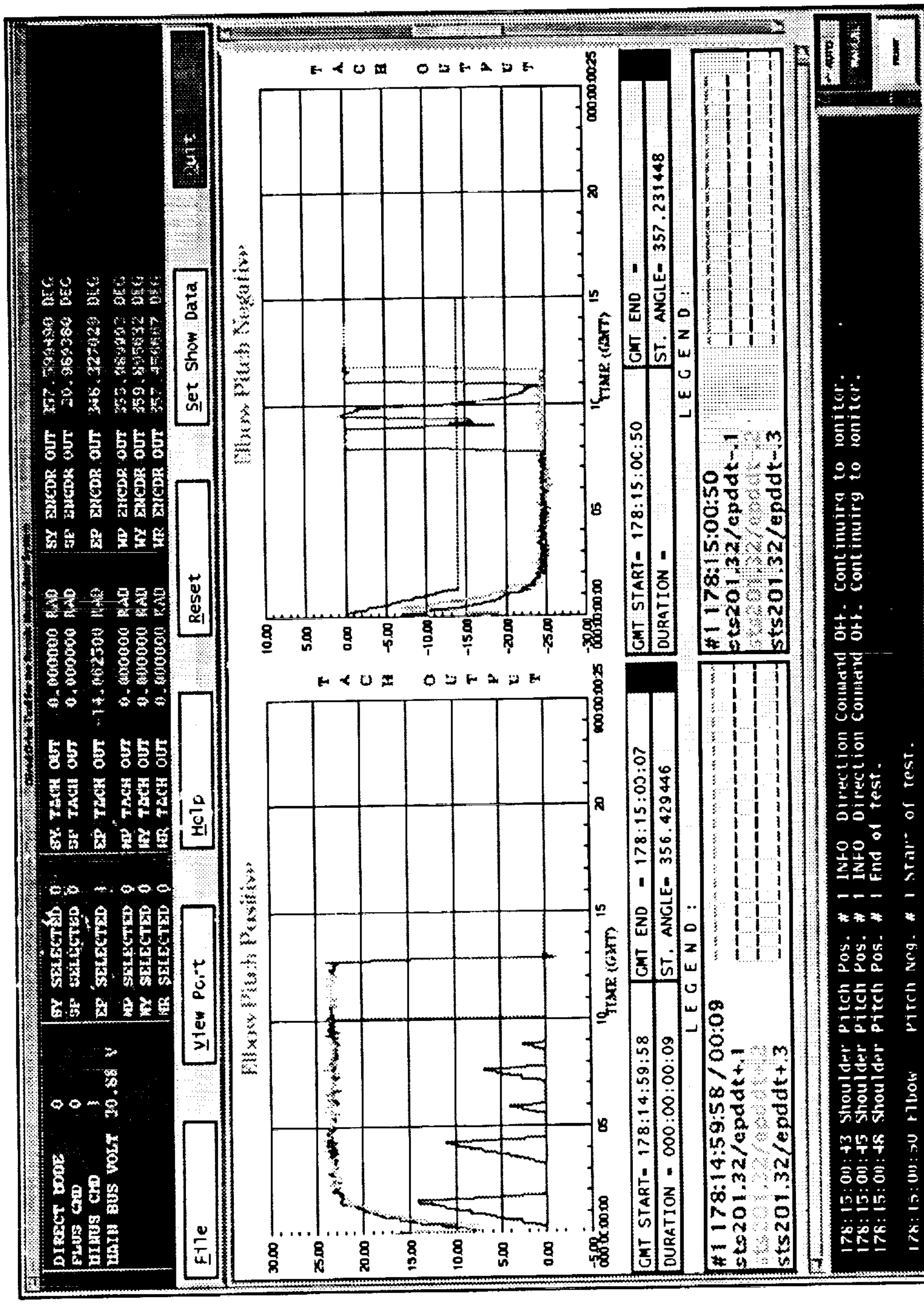

Figure 5: Direct Drive Test Monitor

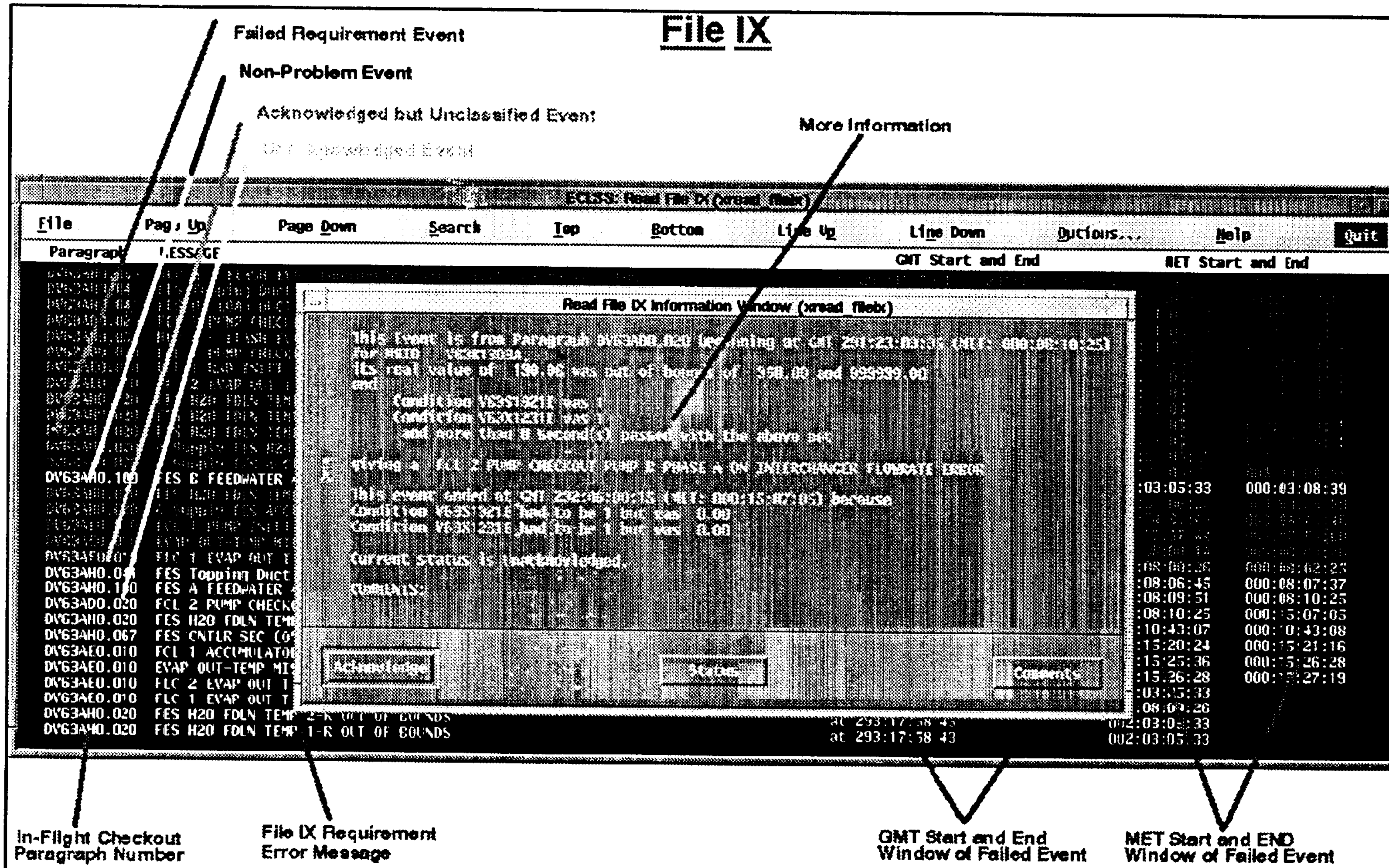


Figure 6: Automated In-flight Check-out Application

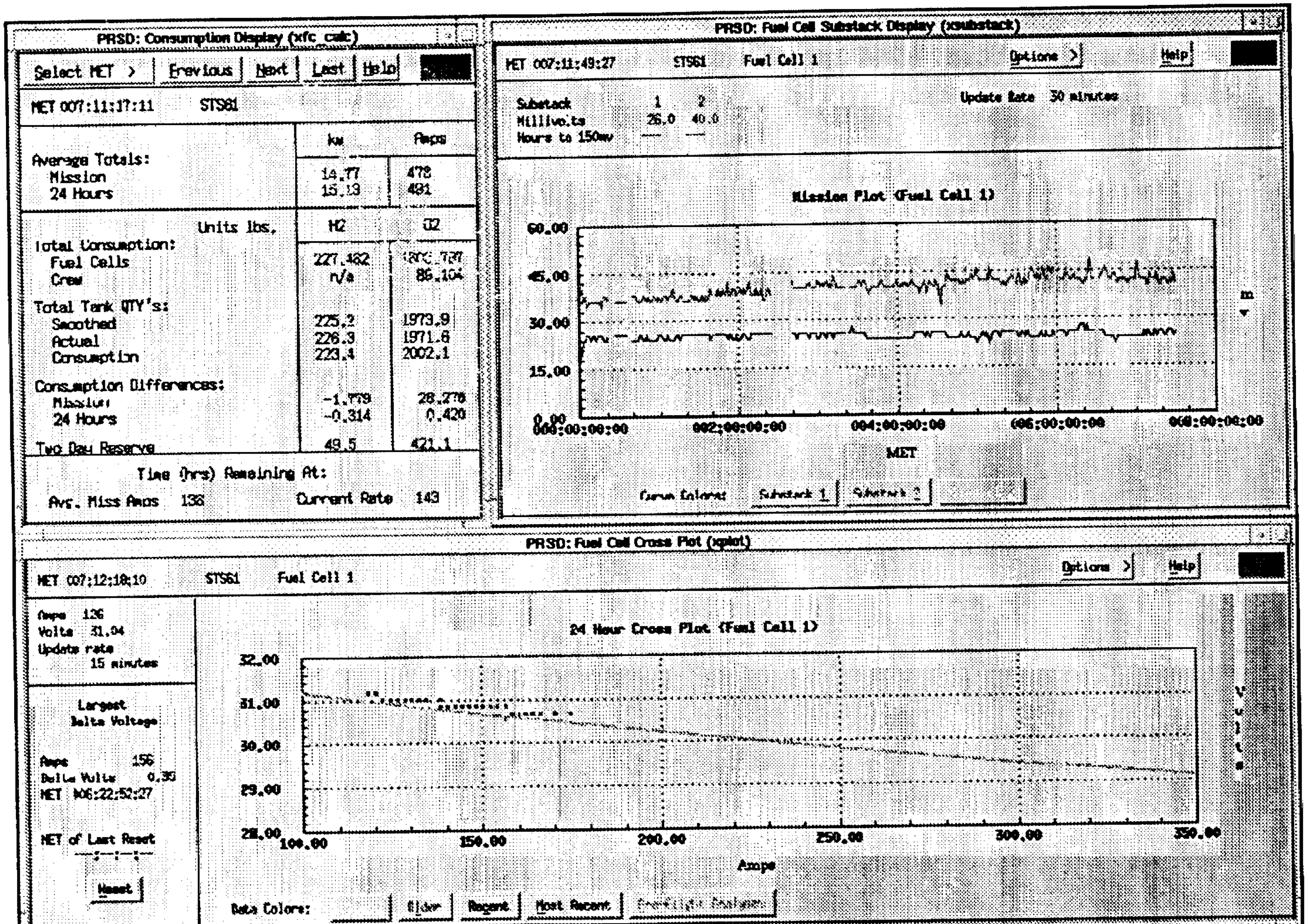


Figure 7: PRSD Trend Analysis Application

burns, on/off cycling of mixer heaters, and tank content stratification and destratification, render the sensor readings unreliable for positively determining quantities. Consequently, before any progress could be made to detect leaks, the first task was to attempt to accurately determine the amount of fuel in the tanks.

The Tank quantity measurements consists of 1) the raw measured total tank quantity of O2 and H2 throughout the flight, and 2) the calculated consumption of O2 and H2 based on the power provided by the fuel cells. If the measured tank quantity starts dropping faster than the calculated consumption there may be reason to suspect a leak in one of the tanks. Compensation for the variability in the measured sensor values is obtained by applying an exponential smoothing algorithm to the sensor readings. The smoothed values are plotted against the calculated consumption. A confidence interval is calculated to provide an indication of how well the calculated value tracks the adjusted measured value. A significant divergence between the two measurements provides some indication of a possible leak that should be investigated further.

The measured tank quantity data contains variability introduced at the sensor. One of these is caused by the heater cycle turning on and off to maintain the tank's pressure at a level such that it will become the active tank. These variations have an oscillation-like characteristic with a period of roughly one to two hours. In order to facilitate the comparison of measured quantity to calculated quantity, an exponential smoothing technique was selected to compensate for these variations. The effect of the exponential smoothing technique is similar to that of a moving average in which the data is weighed heavier toward the leading edge and becomes insignificant in the distant past. Furthermore, since real time telemetry data contains small gaps, a technique which can handle irregularly spaced data and which was published by D.J. Wright [1] was selected. This technique is an extension of a double exponential smoothing adapted to data occurring at irregular time intervals.

Since the reactant tank quantity data has a trend component (the quantity is decreasing), a double exponential smoothing technique is required. Double exponential smoothing separately smooths both the quantity estimate and the trend value. The smoothed quantity estimate is obtained by using the following equation:

$$\hat{\mu}_{n+1} = \hat{\mu}_n + (t_{n+1} - t_n)\hat{\beta}_n$$

where

$\hat{\mu}_n$ = the smoothed quantity estimate at time $t_n$, and

$\hat{\beta}_n$ = the smoothed estimate of the trend at time $t_n$.

The smoothed telemetry quantity measurements may be compared to the quantity estimated based on a separate mathematical model of oxygen and hydrogen consumption. This model uses the voltage and current telemetry data to compute estimated reactant consumption based on the chemical reaction equations for the fuel cell. This consumption estimate contains noise as well and is exponentially smoothed using the same technique used to smooth the telemetry quantity data.

As an aid for the comparison of the two measurements, a confidence level calculation testing that the two slopes are statistically equal is computed. The trend estimates from the mathematical model and the telemetry quantity data are used as input parameters to calculate a t-statistic which represents the probability that the two slopes are equal. This gives a number between 0% and 100% which reflects the degree of certainty that the two slopes are the same. This number is used to alert controllers of anomalous conditions in the fuel cell sub-system.

The next phase of this effort consists of investigating the use of learning algorithms and pattern recognition techniques to identify and classify trends that are reflected in the data.

## FY 94 PLANS

### PRSD Trend Analysis

Work in FY 94 will continue the trend analysis of the PRSD data. Pattern recognition, neural networks, and learning systems will be investigated and applied to this domain. These technologies are being applied to four tasks that have been identified for this area; recognition and prediction of destratification in the oxygen tanks, prediction of stratification in the oxygen tanks and recognition of the stratification signature, oxygen/manifold leak detection and trend analysis, and hydrogen tank/manifold leak detection and trend analysis.

The resulting drastic pressure drop that occurs when oxygen tank contents are destratified, (the remixing of a fluid that is non-uniformly dense (stratified), can be confused with large leaks that would also be indicated by a sharp decline in tank pressure. The ability to determine when and how much of the tank contents become stratified, is an important aspect of PRSD failure analysis. It is also important to know when remixing of the contents has occurred. The extent of existing tank content stratification, and the type of thruster burn that is to occur can be used to predict the probability of destratification, and assess its severity. The signature for an impending destratification is identified by a drop in pressure of as much as 300 psi within 15 minutes, caused by the cooling induced by mixing high density (cool) liquid near the tank walls, with the hot, low density liquid near the tank heater. This may occur during a maneuver after a long period of no acceleration of the vehicle. The severity of the destratification is affected by the distance of the tank form the center of gravity, and the type and direction of the thruster burn. Over 30 recorded instances of destratification will be used to train a system to identify the signature of destratification, anticipate the probability of an impending destratification, and assess the severity of tank destratification.

A slow, gradual decrease in measured tank quantity that would indicate a small leak, can also be the result of content stratification. Another aspect of PRSD trend analysis is to recognize the signature of stratification, and asses to which degree the contents have become stratified. Stratification affects the reliability of the sensor reading, an effect that will also have to be quantified. Numerable data points on stratification are available and will be used to train the system to recognize the stratification signature.

### ATCS Diagnostic System

The Active Thermal Control System consists of several subsystems that together provide cooling for the shuttle. The ATCS Diagnostic System is to be comprised of diagnosis modules for the Flash Evaporator, Ammonia Boiler, Radiator, Freon Coolant Loops.

The first diagnostic system to be developed is that of the FES. The shuttle's Flash Evaporator System (FES) provides heat rejection of shuttle thermal loads. Poor instrumentation and too frequent failures have made this a prime area of investigation for automated diagnosis. The concept for this application was completed in FY93, and development is proceeding with knowledge capture of system expertise, and development of detailed requirements. The FES Diagnostic System will detect, identify, and diagnose failures that could cause the FES to stop working, a condition known as "FES shutdown". Seven categories of failures have been identified that would lead to FES shutdown; 1) restricted Freon flow to the FES, 2) Freon leakage, 3) water spray valve module failure, 4) steam duct failure, 5) accumulator failure, 6) controller

failure, and 7) FES supply water heater failure. The FES diagnostic system will address each class of failures.

## SUMMARY

MIDAS applications have provided engineers and flight controllers at the Johnson Space Center, with software tools that reduce the amount of work involved in supporting missions, and that capture vanishing system expertise. Midas applications off-load to computers and software, the tasks of data gathering, filtering and analysis, perform automated diagnosis of system problems, and allow engineers to quickly and accurately identify and resolve problems. MIDAS applications have positively affected MER mission support by allowing subsystem managers to reduce MER mission support staffing, and by allowing subsystem personnel to share diminishing corporate knowledge and responsibility for their systems. Additionally, MIDAS applications have eliminated tasks previously required of humans, supplied previously unavailable data and analysis, and eliminated the need for some post-mission analysis. Some applications such as the Discrete Monitor and Log, Analog Plots, and File IX applications are re-usable and have been cloned to quickly distribute these capabilities to other systems, thereby increasing the cost effectiveness of development.

MIDAS Applications are written in the "C" programming language using X-windows version 11, Release 5 (X11R5), and the Motif widget set. Also used is Gensym's G2 expert system shell. The applications currently run on the SUN (OS 4.1.3 version of UNIX) Sparc1+ and Sparc2 machines, Masscomp workstations, and DEC-stations.

## ACKNOWLEDGEMENTS

The MIDAS Project is sponsored by the Automation and Robotics Division, Intelligent Systems Branch, with funding from the Orbiter Projects Office, Code VA of the Johnson Space Center. The NASA Project Manager for this task is Ginger L. Pack/ER2. Programming support is performed by the NASA Engineering Support Contractor, Lockheed Engineering and Sciences Corporation (LESC). The programmers for this task are Jane Falgout, Joe Barcio, Bryan Kubena, and Charles Smith, all of LESC. The technical lead is Louis Flores of LESC.

## REFERENCES

1. D. J. Wright, FORECASTING *IRREGURLARLY SPACED DATA: AN EXTENSION OF DOUBLE EXPONENTIAL SMOOTHING*, Comput. & Indus. Engng, Vol. 10. No. 2, pp 135-147, 1986.

2. *KU-BAND RADAR SELF TEST APPLICATION REQUIREMENTS*, April 1993

3. Steve Schnurer, *TREND ANALYSIS: APPLICATION EVALUATION AND RECOMMEDATION*, November 1993